Automatic Detection and Classification of Anti-islamic Web Text-Contents

Rawan Abdullah Alraddadi[1(✉)] and Moulay Ibrahim El-Khalil Ghembaza[2]

[1] Department of Computer Science, College of Computer Science and Engineering, Taibah University, Medina, Saudi Arabia
rawanalradadi3@gmail.com

[2] Department of Computer Science, College of Engineering and Information Technology, Onaizah Colleges, Qassim, Saudi Arabia
mghembaza@oc.edu.sa

Abstract. The aim of this research is to use the sentiment analysis techniques to deal with large dataset corpus, which has been collected, to detect and classify anti-Islamic online contents. Anti-Islamic websites have spread a lot in the last decade causing a lot of hate toward the Muslims communities; there have been many websites that attack Islam and Muslims and insult the Messenger, blessings and peace be upon him. We have gathered our proper dataset from different sources into a large corpus, and we have produced two datasets (balanced and non-balanced) for the English language. The framework of our proposed methodology has been described. Two approaches are used in this framework, the first one is based on supervised Machine Learning (ML) approach using Support Vector Machines (SVM) model as classifier and Term Frequency-Inverse Document Frequency (TF-IDF) as feature extraction; the second one is a hybrid approach combining lexicon-based dictionary and TF-IDF as feature extraction with SVM algorithm. We conducted different experiments and we compared the obtained results. We first use TF-IDF on word level, and then we have improved the model using tri-gram level. The experimental results show that the ML approach is the best approach for both datasets that produces high accuracy of 97% applied on the non-balanced English dataset using SVM with tri-gram level TF-IDF as feature extraction. Additionally, SVM with word-level TF-IDF also provides excellent results regardless of the type of dataset.

Keywords: Web text mining · Text analysis · Text classification · SVM · Sentiment analysis · Fake news · Hate speech · Toxicity detection

1 Introduction

In the last decade, there was a lot of hate in the world toward Islam, Muslims and even some have insult the Prophet Muhammad, peace be upon him. For some people, the

The original version of this chapter was revised: The author's last name and first name order has been corrected as "Ghembaza, Moulay Ibrahim El-Khalil". The correction to this chapter is available at https://doi.org/10.1007/978-3-031-04409-0_33

X. Jiang (Ed.): MLICOM 2021, LNICST 438, pp. 162–181, 2022.
https://doi.org/10.1007/978-3-031-04409-0_16

hate has gone beyond verbal and physical assaults, and some have committed murders and hate crimes against Muslims and Islam. The hate crime against Islam has increased and in the Al-Noor Mosque in New Zealand, 51 people were killed just because they were Muslims [1]. Moreover, one of the recent incidents occurred in September 2020 in France is the insult to the Prophet Muhammad, peace be upon him, by a history teacher named Samuel Patti who showed some caricature that offend the Prophet [2]. Furthermore, Muslims get a lot of abuse and insults through the Internet. In recent times, there have been many websites that attack Islam and Muslims and insult the Messenger, blessings and peace be upon him, and some other websites contain information that promotes hated and terrorism toward Islam and Muslims, and publish misleading, false, and fake news.

The hate toward Islam has increased due to different reasons including the tragedy that took place in America on September 11, 2001, and the terrorist acts that are taking place in the world and the Middle East in the name of Islam [3]. Due to these incidents, the media have focused on Islam and show the terrorist acts that are done by some radicals and claimed that this is what Islam is calling for. Some peoples have believed these claims, which yield some anti-Islamic websites where they express their thoughts and spread hate towards Islam and Muslims.

Anti-Islamic websites have spread a lot in the last decade causing a lot of hate toward the Muslims communities especially people who live in foreign countries or any places containing extremists who are against Islam. Therefore, we need to stop this as it affects many people and gives a bad image for Islam. This can be done by many ways; one of them is detecting and classifying those websites to try limiting their existence. These websites will be gathered into a database to be processed and validated, then analyzed using sentiment analysis techniques. Using this method will allow us to extract the meaning from a large text corpus, which contains opinions, attitude, thoughts and emotions to detect and classify whether the webpage is anti-Islamic or not. Anti-Islamic websites real-time detection will be considered as a new research topic, where it is worth for researchers to take it into consideration since there are about 1.8 billion Muslims in the world[1].

The social impact of this research is attempting to limit the spread of false information about Islam and Muslims by detecting and classifying hostile websites against Islam. This may help to stop the wrong perception about Islam and spreading correct information about this religion to all the humanity, without including any form of extremist views and ideas of some bad people. Therefore, the need of real-time detection of anti-Islamic online content, using machine learning-based sentiment analysis techniques, or any other techniques, is necessary to prevent any terrorist acts towards Islam and Muslims.

To the best of our knowledge, there is no previous research prior to our research that addresses this issue and provides an automatic detection method of anti-Islam online contents. Unfortunately, we did not find papers that discuss this issue; therefore, we will give an overview of the existing detection methods in different related fields like fake news detection, racism and hate speech detection, and toxicity detection.

This research focuses on detecting anti-Islamic websites using machine learning (ML) techniques. Our main contribution is to provide a huge dataset for anti-Islamic

[1] According to Wikipedia, the free encyclopedia, 2021.

websites to help governments and researchers in the future. In addition, we identify the features that can be used for this purpose and develop a system that will classify the webpages based on their contents.

The remainder of the paper is structured as follows: Section two provides the necessary background. In section three, we review some related work. Section four demonstrates our proposed framework along with the data collection and the various stages of our methodology. Section five includes experimental results and discussion. Finally, section six concludes the paper with a summary and future work.

2 Background

2.1 Sentiment Analysis and Classification

Sentiment Analysis or opinion mining is a natural language processing (NLP) technique that deals with a large text corpus, which contains opinions, attitude, thoughts and emotions to measure the polarity (positive, negative or neutral) in a given document. The process consists of different steps including pre-processing of the document; this may include tokenization, stop-words removal, special symbol removal and other pre-processing techniques. Then, extracting features, which is the process of converting the text in the dataset into a feature set that can be used by the classifier. There are different feature extraction techniques including Term Frequency-Inverse Document Frequency (TF-IDF), N-gram, Bag of Words and other techniques. Then, the training process of the ML model involves the learning algorithm with the training data to learn from. The next step is the classification of the sentiment as positive, negative or neutral; and the aggregation of them. Finally, evaluating the performance of the model using one of various measures such as, Cosine Similarity, Jacquard Similarity, Perplexity and Word Error Rate.

The sentiment classification can be done in different methods as shown in Fig. 1, the first group of methods is lexicon-based that uses sentiment lexicons to assign to each word their sentiment orientation (positive or negative). This method is divided into two approaches: dictionary-based and corpus-based. In dictionary-based the classification relies on a predefined dictionary of terms, while corpus-based does not rely on a predefined dictionary, it uses the statistical analysis of the contents of documents [4]. This group of methods has some limitations and drawbacks because sometimes the domain is not considered, also some new vocabulary and informal language are not considered in addition to other issues.

The second group of methods is based on ML techniques, which finds associations between features extracted from documents and sentiments. It has proved that it is very useful in classification [5]. Sentiment Analysis uses ML techniques that learn and improve based on previous experience, in order to help in classification and predictions of sentiments as positive, negative or neutral. ML is usually categorized as supervised, unsupervised and semi-supervised learning algorithms [5].

Supervised ML algorithms use pre-labeled classes to predict future results based on the past data; the algorithm classifies the dataset with the support of trained classifiers. Supervised algorithms are grouped into classification and regression. In classification, the output variable is a category of various classes; whereas in regression, the output

variable is a real value. The algorithms that are most widely used in supervised learning are Linear Regression, Random Forest and Support Vector Machines (SVM).

Unsupervised ML algorithms use unlabeled input data to find the hidden structure or pattern using different algorithms because they do not have a pre-labeled class to be used in the training of classifiers. Unsupervised algorithms are grouped into clustering and association. A clustering is when we need to discover the inherent grouping in the data; whereas an association is when we need to discover rules that describe huge portions of data. The algorithms that are most widely used in unsupervised learning are K-Means and Apriori Algorithms.

Semi-supervised ML algorithms combine both labeled and unlabeled datasets. It combines a small set of labeled data with a huge dataset of unlabeled data to train the classifier. With the help of supervised and unsupervised ML algorithms, we can predict whether or not a giving website contain information that promotes hated and racism toward Islam and Muslims.

The third group of methods uses deep learning techniques consisting of multiple layers with a middle hidden layer to solve complex problems. It is an evolution of ML techniques, where the features are learned and extracted automatically, and provide better performance and accuracy [4]. There are different algorithms that are widely used in deep learning techniques such as Deep Neural Networks (DNN), Convolution Neural Networks (CNN) and Recurrent Neural Networks (RNN) [4].

The fourth group of methods is the methods based on ontology, where they model the concepts and terms in the domain knowledge based on the interest and the relations among these terms. The ontology model consists of entities, objects, properties of objects, and relations between them; as well as the common vocabulary in a domain [6]. The use of ontology helps in making the knowledge easier to understand for the people and the software agents, and also differentiate between words which have the same meaning [7].

Other sentiment analysis methods exist, namely the hybrid method where it combines different approaches to proposed new methods that can optimize the result. This method takes the advantage of the combined approaches to achieve the classification goal with a high accuracy. Some proposed hybrid methods combine lexicon-based approach and ML approach; others combine symbolic approach where it models structured domains and relations among objects, and statistical approach, which can model uncertainty in a robust manner and take advantage of both approaches [8, 9].

2.2 Fake News

Fake news is a long-lasting problem that has existed since the beginning of the printing press in 1493. Fake news is known as news articles that contain false information about a particular subject to mislead the readers to think that the presented information is true. Unfortunately, most of this news are intentionally and verifiably false, and detecting that fake news is considered a hard task due to the speed of spreading this wrong information and the availability of its content, which makes it hard to control [10].

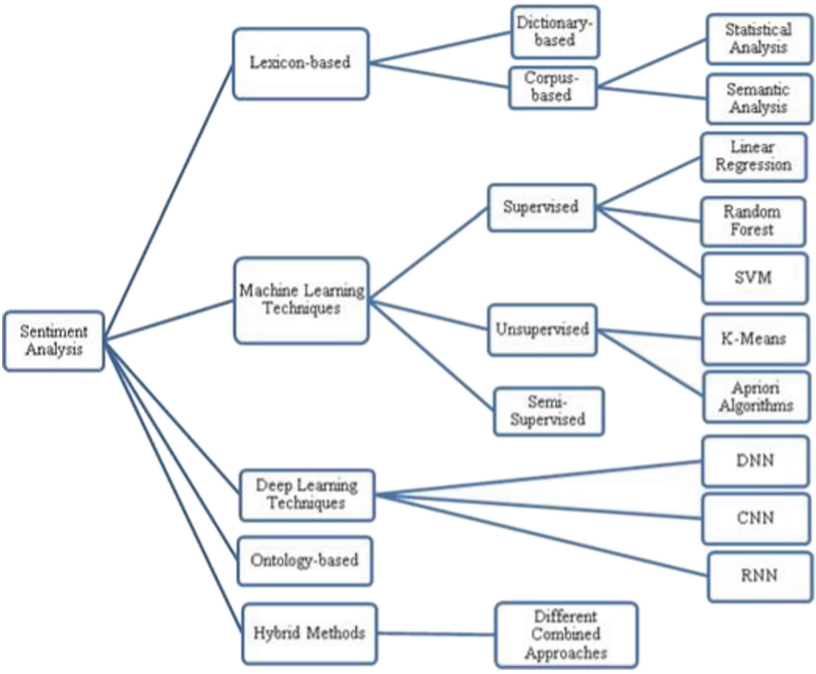

Fig. 1. Sentiment analysis methods.

2.3 Hate Speech

As the number of Internet users increase and the anonymity of their personal information increases, they have the freedom of speech where they can express their thoughts freely, but some of them misuse this by spreading hate and offending others [1, 11]. The number of articles containing hate and racist speeches towards a certain society or type of person has increased. Some jokes and comments can be considered as hate speech as it spreads wrong and hateful information about a particular subject, company or a specific community or even a person. In some counties such as Germany and Iceland, this type of comment and speech are punishable by the law, therefore, detecting and classifying hate crime, racism, hate speech, abuse, harassment, and toxic language online contents are considered important tasks in the NLP field.

3 Related Work

3.1 Fake News Detection

Automatic detection attracts the attention of several researchers; Ghosh and Shah [12] proposed and developed a generalized method based on DNN. They used two databases based on the length and structure of the sentences. Their model starts by classifying the topics using a modular approach in order to detect if the news is fake or true. The model consists of different sub-modules that categorize the instances based on predefined

features including the authenticity of the source, style, natural language features and other features, to predict the credibility of the news. They have combined different techniques from information retrieval, NLP, and deep learning to compare two main sub-modules. The first sub-module uses information retrieval relevant, knowledge base and word-level features; whereas the second sub-module uses a DNN. Their classification model has an accuracy of up to 82.4% by combining the two models. The accuracy of their model is not that good therefore; it needs to be improved [12].

In another research, Barbosa et al. [13] proposed an automatic detection and classification method that uses a Multi-layer Perceptron (MLP) neural networks algorithm called AuFa. They obtained the dataset from a Kaggle repository called "Fake News-Build a system to identify unreliable news". The dataset has 20,800 news but 39 of them are null. The model is divided into three layers: classification, web search and database. In the classification layer, the content of the document goes through a preprocessing process, where the unnecessary words are removed and only the unique and meaningful words stay and turned into lowercase without double spacing. After preprocessing, the data are converted into numeric values to check the credibility of the document and if it is less than 90%, it will move to the second layer. In the web layer, the document is checked using a similarity grouping technique to search for similar results on the web. Whilst, in the database layer, the new data is saved in the database. Comparing their MLP Classifier algorithm with Naïve Bayes, Stochastic Gradient Descent (SGD) Classifier, and SVM algorithms, the results showed that their classifier has a better accuracy with 96.44%, but still needs to be improved [12].

Some research, instead of proposing and developing a new method, they improved an existing method by combining the existing method with new data mining techniques. Rukavitsyn et al. [14] proposed several webpage categorization methods using neighboring webpages. The first method classifies the page based on text data by filtering the page based on "correctly predicted". The second method classifies the page based on two level keywords. The third method is based on neighboring pages, where the main page has links to other subpages depending on their distance; this method is considered complex and produces a low accuracy rate. The last proposed method was the best one, which is a hierarchical classification method; it links different classifiers that are trained with different attributes and algorithms. They compare k-Nearest Neighbors (k-NN), SVM, Logistic Regression, Decision Tree, and Bagging Random Forest, and the results have shown that the SVM model had the highest accuracy. The open issue in this research is how to classify a webpage that doesn't contain text content but only contains images.

Vaibhav et al. [15] proposed a graph neural network-based model because of the diverse interaction between words on different websites containing long text. They used three databases: Satirical and Legitimate News Database (SLN), Random Political News Dataset (RPN) and Labeled Unreliable News Dataset (LUN). They implemented three neural baselines: CNN layer, Long Short-Term Memory layer (LSTM) and Bidirectional Encoder Representations from Transformers (BERT). The contents of the page are represented by a fully connected graph, where the nodes represent the sentences in the page, and the edges between those nodes are similar. Their experiment was done on two types of graph neural networks. The first one is Graph Convolution Network (GCN) where it predicts the relationship between two nodes in the graph [16]; and the second

one is Graph Attention Network (GAT), which is developed to improve the GCN, and it focuses on computing the hidden representations of each node without depending on the graph structure [17]. Their proposed methods outperform the three best baselines.

The research conducted by Ahmed et al. [18] showed that most of the people in the US depend on online news articles, which may contain fake news rather than mainstream media. Therefore, the authors proposed a detection model using word-based n-gram analysis. The purpose of this model is to represent the word in the article and generate features so that it can classify the article. Their dataset was collected by their team, and collected publicly available news articles. They also test their model on the publicly available dataset Horne and Adali[2]. They presented and compared two features selection methods that are Term Frequency (TF) and TF-IDF. TF counts the number of words appearing in the article to calculate the similarity between articles while TF-IDF is based on the importance of the words in the article where it increases whenever it appears in the article. The detection model also uses six machine classification techniques, that is, SGD, SVM, Linear Support Vector Machines (LSVM), Logistic Regression, k-NN, and Decision Tree. The experiment showed that the best performance as a feature extraction technique was TF-IDF, while the best performance as classification techniques was LSVM with an accuracy of 92%.

Also in a newer paper published in 2020, Yazdi et al. [19] used a feature selection method combined with K-means clustering and SVM approaches. They used different datasets with different features to evaluate their method, namely, BuzzFeed News[3] including 1627 papers, BS Detector[4] and LIAR[5] including 12836 brief statements. Their method has four steps; the first one calculates the similarity between all the features in the dataset. The second step categorizes those features into clusters using the k-means clustering method based on similarities. The third step chooses the final features of clusters based on the appropriateness of the features to reduce the size of the dataset. The final step detects fake news using the SVM algorithm. The results have shown that their proposed method has a better classification compared to other methods that use a feature extraction approach [19].

Aphiwongsophon and Chongstitvatana [20] used the normalization rules for cleaning data before using ML methods for classification. They used three ML methods: SVM, Naïve Bayes and Neural networks. Their experiment started by collecting data from Twitter, then applying normalization rules and removing the unnecessary data, and finally classifying the data using ML methods. The experimental results showed that Naïve Bayes has the lowest accuracy between them with 96.08%, whereas SVM and Neural networks have equal accuracy with 99.90%.

Moreover, another research has been done on Twitter by Mahir et al. [21], where they proposed a model for identifying fake news from the tweets. They used Chile earthquake 2010 Datasets consisting of 20,360 Twitter data posts in the dataset. They also compared five ML algorithms: SVM, Naïve Bayes Method, Logistic Regression, and RNN models. The experiment results showed that SVM and Naïve Bayes had a better performance

[2] https://github.com/BenjaminDHorne/fakenewsdata1.
[3] https://github.com/BuzzFeedNews.
[4] https://github.com/thiagovas/bs-detector-dataset.
[5] https://paperswithcode.com/dataset/liar.

with 94% for both of them in terms of F1 score that is the average weight of Precision and Recall.

Another approach has been adopted where the algorithm depend on the subjectivity of the language proposed by Libanio et al. [22]. Their dataset of legitimate news consists of 207,914 articles collected between 2014 to 2017 from two news sites in Brazil, namely Estadão and Folha de São Paulo. They used semantic distances as features between the document, and the five subjectivity lexicons, which are argumentation, presupposition, sentiment, valuation and modalization. Each document will have a vector to calculate the Word Mover's Distance, where it calculates the minimum distance for a word to reach another word in the embedding space for the classification. The results showed that this approach has better results than classical text classification. As well, it is good when the training and testing domains are different [22].

Ozbay and Alatas [23] adapted two metaheuristic optimization algorithms. The first one is the Grey Wolf Optimization (GWO), where it has a candidate solution that is divided into four groups for the problem. The second one is Salp Swarm Optimization (SSO) and has a number of n-dimensional random candidate solutions. The fitness values for these solutions are calculated and the best one is chosen. They used three datasets: BuzzFeed Political News, Random Political News and Liar Benchmark. Their approach consists of three steps, the first one was data preprocessing, the next step was adapting the GWO and SSO to construct a model for fake news detection, and the last step was testing the proposed model. Their experiment has been constructed using three datasets and compared with seven supervised artificial intelligence algorithms. The results have shown that GWO has the highest accuracy with 96.5% between them all, but SSO has a better performance in terms of precision with 100% between them all, within two out of three datasets. However, the algorithms need to be improved.

3.2 Hate Speech Detection

Asmi and Sanaj [11] proposed a word embedding approach and deep learning techniques to automatically classify toxic speech. They combined three feature extraction methods: TF-IDF, fastText Embedding and BERT Embedding. They later extracted the feature from the converted text into numerical form. They used a DNN classifier including CNN and BiLSTM. Perform their classification on binary and multi-class corpus extracted from Twitter and Facebook. In addition, they gave a brief review of the different existing techniques that detect different kinds of speech. Unfortunately, they did not give enough details about the data cleaning and data collection process or their implementation.

In paper [24], D'Sa et al. proposed the same approach as mentioned in paper [11] where they used word embedding representations and deep learning techniques. Their proposed approach used in two ways feature-based and fine-tuning approaches. They used an available dataset that consisted of 24883 tweets and annotated by CrowdFlower, which is the leader in enterprise crowdsourcing, that provide different services for companies. Their pre-processing techniques include removing numbers and special characters except some characters like exclamation mark, question mark and other characters. They also removed the user names and any word connected to the symbol @, and they removed "RT" which refers to re-tweet. Then, they split the hash tags into multiple words. Moreover, they used fastText embedding and BERT embedding as feature extraction. In

addition, they performed their classification on binary and multi-class corpus extracted from Twitter only. The difference compared with [11] is that they used two approaches, in the first approach, they extracted the word embedding and then used a DNN classifier, and in the second approach, they performed fine-tuning of the pre-trained BERT model. Their experimental results showed that BERT fine-tuning outperformed feature-based approaches, where the first one can detect the hate speech up to 53% whereas the feature-based detects only 31% [24].

Fortuna et al. [25] discussed in depth the publicly available datasets that have been introduced to the field of hate speech classification. The authors analyzed six different publicly available datasets based on their similarity and compatibility, and have clarified the categories of the datasets. They referred to the datasets as Waseem, Davidson, Amievalita, Hateval, TRAC, and Toxkaggle. They have conducted two different experiments. In the first experiment, the pre-processing techniques used are to lowercase all words in the corpus, removing IPs, hashtags and usernames. They also removed all the stop-words in the corpus. They trained the word embedding using fastText word vectors and pertained embedding. They compared the classes based on the similarity to other classes in the dataset and their homogeneity. In the second experiment, they used the Perspective API Toxicity classifier, where the classifier calculated the score of the class, which is between 0 to 1. The classifier uses trained CNNs with GloVe word embedding fine-tuned. They performed binary classification and the evaluation is done based on how well the classification algorithm can detect the harmful messages from non-harmful messages. To evaluate the performance of the classifier on the different datasets, they used the F1 metric. The results of their first experiment showed that hate speech categories are very close and similar as well as the aggression categories. However, the categories that contain pejorative speech like toxicity are not related to each other. The results of their second experiment showed that using generic categories data samples or inconsistent annotation can cause a variety and divergence of the classifier performance [25].

4 Proposed Methodology

The main focus in this research is to use sentiment analysis techniques to deal with large dataset corpus which was collected to detect and classify anti-Islamic webpages. The gathered data can be from texts like books, articles, journals, newspapers, and magazines or via oral such as interviews and speeches, to understand the language and predict the meaning of the text or audio-visual materials. We mainly analyzed written text rather than oral or visual materials. Most of the data we have collected were from articles, journals and some of them are from personal blogs. These data are collected and arranged to create a clean database that has a huge amount of data about anti-Islamic websites to be used in the model that we are developing. Figure 2 shows the framework of our proposed methodology, which consists of five stages.

The first stage is data collection, where we collect the data from different sources into a large corpus, and then produce two datasets (balanced and non-balanced datasets). The second stage is data pre-processing, where we prepare the data by applying some techniques such as normalization, stop-words removal, steaming and lemmatization. The third stage is the stage of selecting the features to be used in the next stage. The fourth

stage is the process of training the ML models, where we provide the ML algorithm with the training data to learn from. The last stage is the evaluation of our ML models.

The models used in this research are based on supervised ML approach using SVM algorithm, and hybrid approach combining dictionary-based with SVM algorithm.

In the first model, we used only TF-IDF as feature extraction technique with SVM as classifiers. For the second model, we implemented a hybrid method that uses a lexicon-based dictionary from the Natural Language Toolkit (NLTK) called Valence Aware Dictionary for sEntiment Reasoning (VADER) [26], to analyze the word sentiment meaning. VADER is a simple rule-based model that contains 7517 words and emoticons with their own sentiment polarity for analysis. It was validated by 10 different independent human judges.

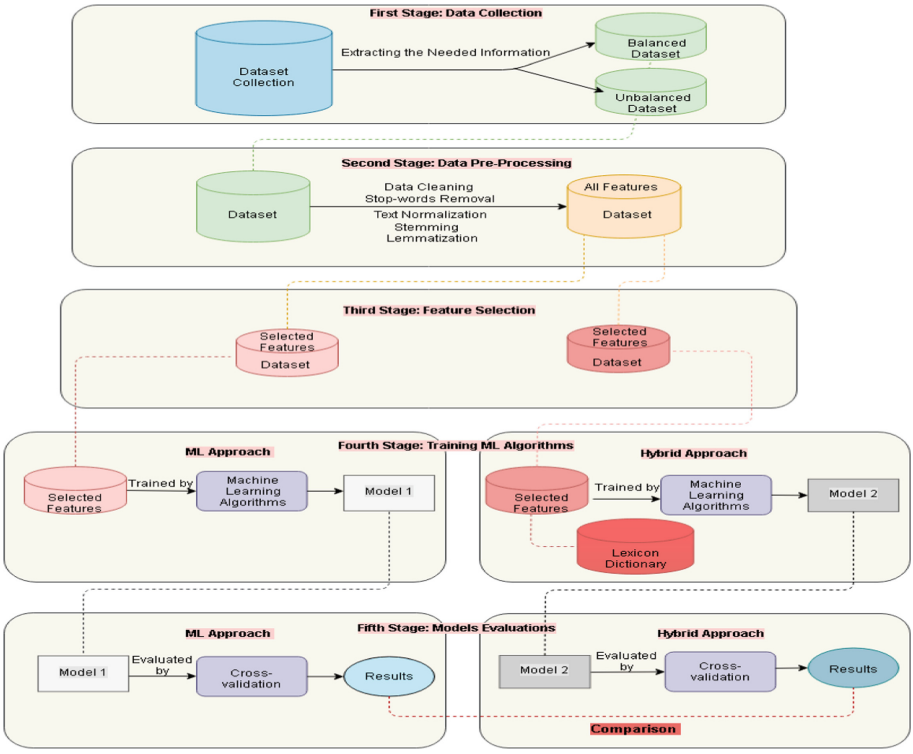

Fig. 2. Framework of the proposed methodology.

4.1 Data Collection and Annotations

We target to create a general benchmark dataset that contains a huge dataset for anti-Islamic websites. The dataset for this research consisted of anti-Islamic websites as well non anti-Islamic websites. The collection of data was mainly done through qualitative data collection methods. The collected data were from articles, journals and some of them are from personal blogs. The main reason for choosing these types of data is because

we are interested in the formal English language used to write academic content and not informal English used in social media. We gathered these data from the Internet using Yahoo and Google search engines. The collection of data started from July 2020 until the end of February 2021.

At first, the data was manually collected, and then we switched to a web-scraping tool called Octoparse. This tool takes the URL of the webpage we want to extract data from, and then we select the target data to be extracted and run the scraping to get the data as CSV, Excel, Application Programming Interface (API), or save them to a database.

The collected data are in English language. The main keywords used in collecting data were: Anti-Islam, Anti-Muslims, Islamophobia, Cyber-Islamophobia, Islamization, Eurabia, Racism, and Islam are false. These keywords help to find the desired content and to decrease the amount of search, due to the enormous quantity of articles that talk about Islam in good or in bad ways. We have two datasets, the balanced dataset that consists of 2092 articles and the non-balanced one, which is made up of long articles containing 2711 articles, and 640 words per article on average. The maximum number of words for the anti-Islamic articles is 3090, whereas for the non anti-Islamic articles is 3281.

After collecting data, we have organized them into an excel spreadsheet, containing the URL, the title, the content, the date, and labeled them as an anti-Islamic webpage or not. We have faced some challenges during the process of collecting data, one of these challenges was that most of the webpages containing extremist ideas or false information about Islam were blocked and we were not able to reach them from Saudi Arabia.

4.2 Data Preparation and Preprocessing

We started by cleaning the datasets. We completed the incomplete date within the dataset. In addition, we found some inconsistent data in the date column; therefore, we converted all dates to "day-month-year" format. Moreover, we removed all duplicated data in the dataset to result in a clean and consistent dataset. Then, we loaded the dataset from the excel file into the Jupyter Notebook in order to prepare the data. In the next step, we perform a sequence of procedures to standardize textual data in a way we could use it.

We perform some pre-processing techniques for the dataset leading to the meaningful information in the text without the unnecessary one. Pre-processing techniques include removing punctuation, removing whitespaces and replacing any phone number with the word phone number, any email address with the word email address. Also, we changed all the words in the text to lowercase. In addition, some of the techniques used are stop-words removal. Stop-words are words in the language that don't have meaning. Removing all stop-words in each article, this will reduce the corpus and speed up the classification process. This method is simple and produces quick and accurate results.

Furthermore, we perform a tokenization, which is the process of dividing the text into a list of tokens that can be either sentences or individual words.

Stemming and lemmatization are a little bit similar; there is not that much difference between them. Using stemming enabled us to save a lot of time as stemming returned the words into their original form, and therefore, a group of words can be reduced to the same stem. Removing word suffixes and/or prefixes can result in a word which is not actually a word in the language.

For the English stemming, we have used Porter stemmer, which was developed in 1979; and then included in the NLTK library; the algorithm removes suffixes to produce the original form of the word. Moreover, for the English lemmatization, we used WordNet Lemmatizer to group different forms of the word to the basic form to be analyzed as the same form. Lemmatization consider the context of the words where it groups the word, which has similar meaning to one word; whereas stemming does not consider the context of the words. Lemmatization can produce accurate results but it require more computation compared to stemming.

4.3 Feature Selection

For feature extraction, we used TF-IDF weighting in both models (ML and hybrid). TF-IDF shows the frequency of a word in the dataset which means that this word has strong sentiment. The value of a word increases with the count but decreases with the frequency of the word in the dataset. In this method, the document contains scores for each word instead of just 0 and 1. The scores can be produced by multiplying the term frequency and inverse document frequency. The term frequency is the frequency of a word in the dataset [27].

Term frequency is calculated by the following equation:

$$TF(word, doc) = \frac{FrequencyOfWord}{NumberOfwords} \tag{1}$$

Inverse document frequency is calculated by the following equation:

$$IDF(word) = log_e\left(\frac{NumberOfDoc}{NumberOfDocWithWord}\right) \tag{2}$$

Therefore, the TF-IDF will be calculated using the following equation:

$$TF - IDF = TF(word, doc) * IDF(word) \tag{3}$$

We have used TF-IDF with word level, where we consider the frequency of a single word in the dataset. Moreover, we have also used TF-IDF with N-gram, which is a model that depends on the sequence of words with a predefined length N to predict the next word. N-gram is another feature extraction technique that is well known used in NLP for language modeling also. There are different popular models for N-gram, the most widely used are word-based and character-based. In our experiment, we used tri-gram, where we consider the frequency of three words in the dataset.

For the second model (hybrid approach), we have used the "SentimentIntensityAnalyzer" from NLTK to categorize each article through VADER sentiment lexicon as positive, negative or neutral using the polarity scores method to get the sentiment for each article. Each word is assigned with its score, and then we used them to conclude the sentiment score for the entire article. We consider the sentiment score results with compound values greater than (0.1) as positive and less than (-0.1) as negative. After that, we used TF-IDF to calculate the weight for each article based on the result from the lexicon. This final result is used as the training data and then fed to our classifier to detect the anti-Islamic content in websites.

4.4 Training and Classification

To achieve our goal in detecting and classifying the anti-Islamic content, we used the SVM classifier for defining the ML model. SVM is a supervised learning algorithm used for classification; the task of the algorithm is to determine which category a new data belongs to based on certain features in the dataset. We choose this algorithm because it is the most suitable algorithm for our dataset as it is less than 100K data. Moreover, this algorithm requires less data for the training to achieve accurate results and produce these results faster than other algorithms.

The dataset is divided into: training, and testing sets. We used the training data to train our model. Furthermore, we used the testing data to confirm that the trained model performs well for the hidden data. We split the data into 70% for training data and 30% for testing data, which will be used in the end when the train of the model is completed.

5 Experimental Results and Models Evaluation

We perform the experiments using 10-fold cross-validation to overcome the overfitting problem in the dataset especially in the non-balanced dataset where a set of partitions for training and testing are used to produce k-classification models. The following metrics are used to evaluate the performance of our model: namely, Confusion Matrix, Precision, Recall, F1 Score, and Accuracy.

Confusion Matrix summarizes the number of correct and incorrect predictions for each class. In the confusion matrix, the row is the actual class, while the column is the predicted class. The four measures, i.e. True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) are used to calculate the following Eqs. (4, 5, 6, and 7) in order to obtain the precision, recall, F1 score, and accuracy:

$$Precision = \frac{TruePositive}{(TruePositive + FalsePositive)} \tag{4}$$

$$Recall = \frac{TruePositive}{(TruePositive + FalseNegative)} \tag{5}$$

$$F1score = \frac{(2 * Precision * Recall)}{(Precision + Recall)} \tag{6}$$

$$Accuracy = \frac{(TruePositive + TrueNegative)}{(TruePositive + TrueNegative + FalsePositive + FalseNegative)} \tag{7}$$

5.1 TF-IDF Vector as Feature

We have used TF-IDF as a feature in our models. We first use it on word level, where it calculates the TF-IDF for each word in the document (article in our case). Then, we have improved the model using tri-gram level, where it calculates the TF-IDF for each three words in the document.

5.1.1 Word Level TF-IDF

Figure 3 shows the confusion matrix using the ML model with a non-balanced dataset, whereas Fig. 4 shows the confusion matrix using a hybrid model with non-balanced dataset. When we use TF-IDF, the True Positive (TP) which is the number of predictions is 699 for the ML model, which is better than 619 for the hybrid model. For the True Negative (TN), which is the correct prediction for the class, the ML model produces 1134, while the hybrid model achieves 1012 correct predictions. For the False Positive (FP), the number of predictions in the ML model is 30, whereas in the hybrid model is 99. For the False Negative (FN), which is the false prediction for the class, the ML model produces 34, but the hybrid model achieves 114 correct predictions.

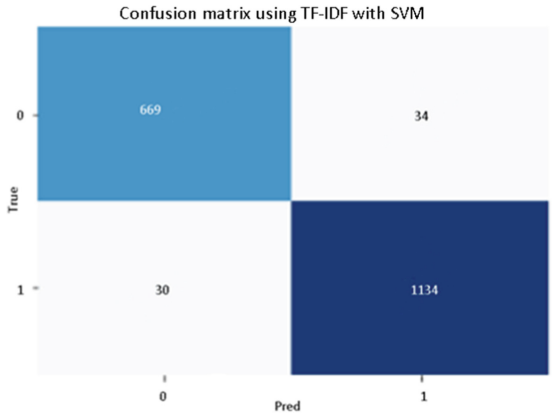

Fig. 3. Confusion matrix using ML model and non-balanced dataset.

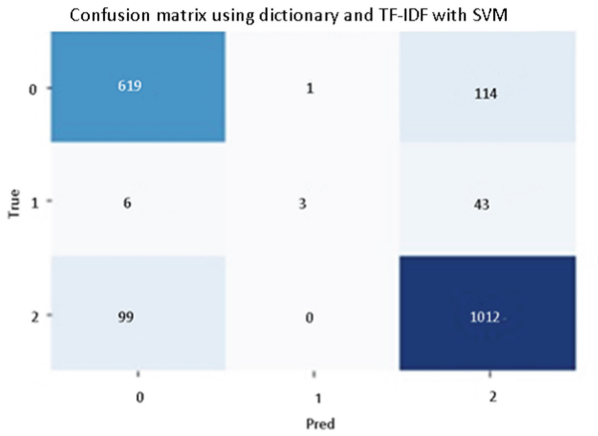

Fig. 4. Confusion matrix using hybrid model and non-balanced dataset.

Table 1 lists the different results between using TF-IDF with SVM (for ML model) and using dictionary and TF-IDF with SVM (for the hybrid model) on a non-balanced dataset. The experimental results shown in Table 1 indicate that the recall and accuracy difference between the two models is close. However, for the two others measurements: Precision and F1 score, the differences are very huge. In Table 2, the accuracy difference between the two models is close. However, for the rest of the measurements, the differences are very huge.

Table 1. Results for word level on non-balanced dataset.

Non-balanced dataset	Precision	Recall	F1 score	Accuracy
Hybrid model	60%	82%	61%	86%
ML model	96%	94%	95%	96%

Table 2. Results for word level on balanced dataset.

Balanced dataset	Precision	Recall	F1 score	Accuracy
Hybrid model	66%	63%	64%	82%
ML model	96%	96%	96%	96%

Table 3 and Table 4 list the precision, recall, and F1 score for the negative articles on the non-balanced and balanced datasets respectively.

Table 3. Results for word level on non-balanced dataset (negative articles only).

Non balanced dataset	Precision	Recall	F1 score
Hybrid model for negative articles	91%	87%	89%
ML model for negative articles	96%	98%	97%

Table 4. Results for word level on balanced dataset (negative articles only).

Balanced dataset	Precision	Recall	F1 score
Hybrid model for negative articles	48%	84%	84%
ML model for negative articles	96%	95%	96%

Consequently, the experimental results show that for our balanced and non-balanced datasets, the best algorithm that produces high accuracy is SVM with word level TF-IDF as feature extraction (see Fig. 5).

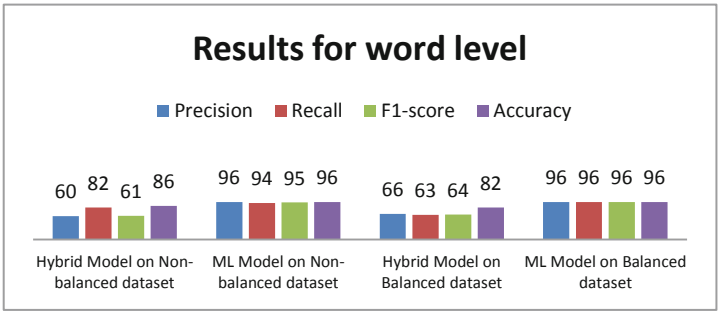

Fig. 5. Results for word level on the two models.

5.1.2 N-gram Level TF-IDF

When we use tri-gram for the hybrid model with SVM, the result has not been modified in all metrics even in the confusion matrix. For the ML model with TF-IDF, the False Positive (FP) becomes 34 and the False Negative (FN) becomes 30. The True Positive (TP) and True Negative (TN) stayed the same. For the negative articles in the ML model, the recall has increased but the precision decreases by one, and the F1 score stays the same.

Table 5 and Table 6 list the different results between using N-gram on the training models with non-balanced and balanced datasets.

Table 5. Results for tri-gram on a non-balanced dataset.

Non-balanced dataset	Precision	Recall	F1 score	Accuracy
Hybrid model	60%	82%	61%	86%
ML model	96%	96%	96%	97%

Table 6. Results for tri-gram on a balanced dataset.

Balanced dataset	Precision	Recall	F1 score	Accuracy
Hybrid model	63%	78%	66%	82%
ML model	79%	85%	78%	85%

Table 7 and Table 8 list the precision, recall and F1 score for the negative articles on the non-balanced and balanced datasets respectively.

Table 7. Results for the negative articles with tri-gram on a non-balanced dataset.

Non-balanced dataset	Precision	Recall	F1 score
Hybrid model for negative articles	91%	87%	89%
ML model for negative articles	97%	97%	97%

Table 8. Results for the negative articles with tri-gram on a balanced dataset.

Balanced dataset	Precision	Recall	F1 score
Hybrid model for negative articles	85%	87%	86%
ML model for negative articles	58%	100%	73%

The experimental results show that for our balanced and non-balanced datasets, the best algorithm that produces high accuracy is SVM with tri-gram level TF-IDF as feature extraction (see Fig. 6).

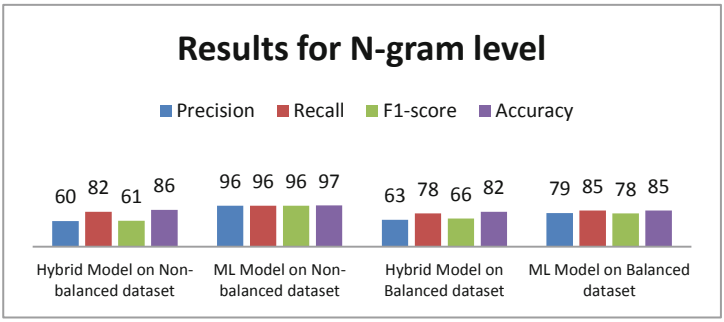

Fig. 6. Results for N-gram level on the two models.

We conducted our experiments on a DELL laptop. The processor is Intel(R) Core (TM) i7-4510U CPU @ 2.60 GHz, the RAM is 8.00 GB, on a 64-bit Operating System. For the non-balanced dataset, the training time for SVM with TF-IDF as feature extraction was 0.096946 s, and the prediction time was 0.001999 s. While the training time for SVM with dictionary and TF-IDF as feature extraction was 0.287853 s, and the prediction time was 0.003998 s. For the balanced dataset, the training time for SVM with TF-IDF as feature extraction was 0.110935 s, and the prediction time was 0.002004 s. While the training time for SVM with dictionary and TF-IDF as feature extraction was 0.660602 s, and the prediction time was 0.003999 s.

6 Conclusion and Future Work

In this research, we have proposed an automatic detection and classification of anti-Islamic websites using sentiment analysis techniques. These websites are considered kind of toxic online contents that encourage spreading hate speech toward Islam and Muslims. Our first objective was to collect our proper dataset and use it to detect and classify the anti-Islamic webpages, to identify the features that can be used in this issue and to create a general benchmark dataset containing a huge amount of data for anti-Islamic and non anti-Islamic websites to help the researchers having and using such corpus.

Some of the limitations that we faced during the process of achieving our goals were the absence of a database that contains anti-Islamic websites neither in English nor in Arabic or other languages. In addition, we have faced some challenges when we were collecting the data, one of the challenges was that a number of webpages that contain extremist ideas or false information about Islam were blocked, and we were not able to reach those webpages from Saudi Arabia. This problem slowed the process of collecting the data and made it harder to find different webpages that contain this kind of information.

We have described the framework of our proposed methodology, which consists of five stages, namely, the data collection, the data pre-processing, the features selection, the training process, and the evaluation of our models. The models used in this research are based on supervised ML approach using SVM algorithm, and hybrid approach combining dictionary-based with SVM algorithm.

The experimental results show that for our datasets, the best algorithm that produces high accuracy with 97% is SVM as classifier with tri-gram level TF-IDF as feature extraction. Additionally, SVM with word-level TF-IDF also provides excellent results regardless of the type of dataset (balanced or non-balanced). This confirm that the SVM algorithm produces high accuracy compared to other algorithms and also it can learn the parameters and produces good results even when the training data are small and this is in a fast training time.

In the future, we continue adding more data to our datasets. We propose to implement a translation-based approach to deal with non English contents such as Arabic or French anti-Islamic text contents; and compare it with standard approaches. Moreover, we will introduce some concepts using NLP techniques to study the effect of a semantic analysis approach and to deal with Arabic texts as well.

Another track to search is exploring different social media to collect data and compare their contents with our dataset, and discover what the experiment's results will show.

References

1. Christchurch shootings: The people killed as they prayed - BBC News. https://www.bbc.com/news/world-asia-47593693. Accessed 26 Jan 2021
2. France Muhammad cartoon row: What you need to know | News | DW | 27.10.2020. https://www.dw.com/en/france-muhammad-cartoon-row-what-you-need-to-know/a-55409316. Accessed 26 Jan 2021

3. Kavakli, K.C., Kuhn, P.M.: Dangerous contenders: election monitors, Islamic opposition parties, and terrorism. Int. Organ. **74**(1), 145–164 (2020)
4. Dang, N.C., Moreno-García, M.N., de la Prieta, F.: Sentiment analysis based on deep learning: a comparative study, arXiv (2020)
5. Becker, K., Harb, J.G., Ebeling, R.: Exploring deep learning for the analysis of emotional reactions to terrorist events on Twitter. J. Inf. Data Manag. **10**(2), 97–115 (2019)
6. Yaakub, M.R., Latiffi, M.I.A., Zaabar, L.S.: A review on sentiment analysis techniques and applications. IOP Conf. Ser.: Mater. Sci. Eng. **551**(1), 012070 (2019)
7. Thakor, P., Sasi, S.: Ontology-based sentiment analysis process for social media content. Procedia Comput. Sci. **53**(1), 199–207 (2015)
8. Nandi, V., Agrawal, S.: Political sentiment analysis using hybrid approach. Int. Res. J. Eng. Technol. **3**(5), 1621–1627 (2016)
9. Alrefai, M., Faris, H., Aljarah, I.: Sentiment analysis for Arabic language: a brief survey of approaches and techniques. arXiv (2018)
10. Shu, K., Sliva, A., Wang, S., Tang, J., Liu, H.: Fake news detection on social media: a data mining perspective. arXiv, vol. 19, no. 1, pp. 22–36 (2017)
11. Asmi, P., Sanaj, M.S.: Online Toxic Speech : Automatic Detection Methods and Techniques, vol. 6, no. 6, pp. 3353–3356 (2020)
12. Ghosh, S., Shah, C.: Toward automatic fake news classification. In: Proceedings of 52nd Hawaii International Conference on System Sciences, vol. 6, pp. 2254–2263 (2019)
13. Barbosa, V., de Oliveira, C., Braga, R.B.: AuFa-automatic detection and classification of fake news using neural networks. In: 8th International Workshop on ADVANCEs in ICT Infrastructures and Services (ADVANCE 2020), Cancún, Mexico, pp. 1–8, January 2020
14. Rukavitsyn, A.N., Kupriyanov, M.S., Shorov, A.V., Petukhov, I.V.: Investigation of website classification methods based on data mining techniques. In: 2016 XIX IEEE International Conference on Soft Computing and Measurements (SCM), St. Petersburg, Russia, pp. 333–336 (2016)
15. Vaibhav, V., Mandyam, R., Hovy, E.: Do Sentence Interactions Matter? Leveraging Sentence Level Representations for Fake News Classification, pp. 134–139 (2019)
16. Wang, Z., Zheng, L., Li, Y., Wang, S.: Linkage based face clustering via graph convolution network. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2019-June, no. c, pp. 1117–1125 (2019)
17. Veličković, P., Casanova, A., Liò, P., Cucurull, G., Romero, A., Bengio, Y.: Graph attention networks. In: 6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings, pp. 1–12 (2018)
18. Ahmed, H., Traore, I., Saad, S.: Detection of online fake news using n-gram analysis and machine learning techniques. In: Traore, I., Woungang, I., Awad, A. (eds.) ISDDC 2017. LNCS, vol. 10618, pp. 127–138. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69155-8_9
19. Yazdi, K.M., Yazdi, A.M., Khodayi, S., Hou, J., Zhou, W., Saedy, S.: Improving fake news detection using k-means and support vector machine approaches. Int. J. Electron. Commun. Eng. **14**(2), 38–42 (2020)
20. Aphiwongsophon, S., Chongstitvatana, P.: Detecting fake news with machine learning method, pp. 528–531 (2018)
21. Mahir, E.M., Akhter, S., Huq, M.R.: Detecting fake news using machine learning and deep learning algorithms, pp. 1–4 (2019)
22. Libanio, C., Jeronimo, M., Campelo, C.E.C., Veloso, A., Sales, A.: Fake News Classification Based on Subjective Language (2019)
23. Ozbay, F.A., Alatas, B.: A novel approach for detection of fake news on social media using metaheuristic optimization algorithms. Elektron. ir Elektrotechnika **25**(4), 62–67 (2019)

24. D'Sa, A.G., Illina, I., Fohr, D.: BERT and fasttext embeddings for automatic detection of toxic speech. In: Proceedings of 2020 International Multi-Conference on Organization of Knowledge and Advanced Technologies, OCTA 2020 (2020)
25. Fortuna, P., Soler, J., Wanner, L.: Toxic, hateful, offensive or abusive? What are we really classifying? An empirical analysis of hate speech datasets, no. May, pp. 6786–6794 (2020)
26. Hutto, C.J., Gilbert, E.: VADER : a parsimonious rule-based model for sentiment analysis of social media text, pp. 216–225 (2014)
27. Das, B., Chakraborty, S.: An improved text sentiment classification model using TF-IDF and next word negation, arXiv (2018)